

COMPOSITE PICTURE GENERATING METHOD

BACKGROUND OF THE INVENTION

The present invention relates to a composite picture generating method of superimposing an image over a moving picture by using a computer, and more 5 specifically to an image synthesizing method suited for synthesizing an arbitrary image moving over a moving picture and a recording medium storing an image synthesizing program.

A commonly used method for superimposing an 10 image over a moving picture generally involves specifying positions on one or more frames at which the image is to be introduced. The area in which the image is to be fitted is generally specified by setting its boundary. An example method of automatically setting a boundary to 15 determine an image synthesizing area in a background moving picture is described in JP-A-7-262410. This method extracts a boundary line in a real video footage, specifies with a pointing device a rough area where a user wants an image to be inserted and determines an 20 exact area of image insertion. In the area thus determined a desired image is superimposed. Once the area is determined, the image can be superimposed over that area without needing to specify the image superimposing position for each frame.

SUMMARY OF THE INVENTION

In the conventional technology described above, when an image to be superimposed is an object that moves in a natural and smooth locus (for example, a bird flying 5 in the sky or a goldfish in a bowl), the method that specifies the superimposing position in one or more frames has a drawback that the motion of the object does not look natural. To make the object look natural in its motion requires detailed calculations to interpolate the 10 frames.

With another method that automatically sets a boundary to determine an area (range) in which the image is to be superimposed, locations other than those the user wants to specify as boundaries are often recognized 15 as boundaries. This method requires high precision in the extraction of the boundary and takes time when an apparatus with a low processing power, such as personal computer for consumer, is used.

An object of the present invention is to 20 provide a composite picture generating method that can easily synthesize an image that moves with a natural motion over a background moving picture and to provide a recording medium storing a composite picture generating program.

25 Another object of the present invention is to provide a composite picture generating method that supports automatically extracting from the background moving picture a movement boundary of an object to be

superimposed over the moving picture.

According to one aspect of the invention there is provided a composite picture generating method which comprises the steps of: specifying a locus of motion of the image while replaying the moving picture; and synthesizing the image moving along the specified locus over the moving picture. This allows for image synthesizing wherein the movement of an object superimposed over the moving picture looks natural.

According to another aspect of the invention there is provided a composite picture generating method which comprises the steps of: specifying a boundary area in the moving picture; and modifying the locus of motion of the image to meet the specified boundary area condition. Further, the method includes the steps of: specifying information for identifying the boundary area, or area identification information, along with the boundary area; based on the area identification information, determining an actual boundary area in the moving picture being replayed; and modifying an image superimposing position. This makes the creation of an animation of an object moving in desired locations easier than when the conventional procedure is used that automatically generates a boundary to determine the area of movement of the object.

Other objects, features and advantages of the present invention will become apparent from the detailed description of the embodiments of the invention taken in

DRAFTING STANDARDS

conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram showing an example configuration of a computer system used in the present 5 invention.

Figure 2 is an example view of a composite picture generating program used in the embodiment.

Figure 3 is a flow chart showing overall processing of a first embodiment of the invention.

10 Figure 4 is a flow chart showing processing for specifying and determining a locus of an image in the first embodiment of the invention.

Figure 5 is an example showing how an image is drawn in the first embodiment of the invention.

15 Figure 6 is an example showing how a locus of motion of the image is drawn in the first embodiment of the invention.

Figure 7 is an example method of modifying the locus of motion of the image in the first embodiment of 20 the invention.

Figures 8A to 8C are tables showing a data structure of the locus of motion of the image in the first embodiment of the invention.

Figure 9 is a flow chart showing overall 25 processing of a second embodiment of the invention.

Figure 10 is a flow chart showing area specification processing in the second embodiment of the

invention.

Figure 11 is an example diagram showing how an area is specified in the second embodiment of the invention.

5 Figure 12 is a flow chart showing processing that limits the locus of motion of the image to within the area in the second embodiment of the invention.

Figure 13 is an example locus of motion of the image in the second embodiment of the invention.

10 Figure 14 is a table showing boundary line data in the second embodiment of the invention.

Figure 15 is a table showing position data of a locus of motion of the image in the second embodiment.

15 Figure 16 is a flow chart showing processing that modifies a locus of motion of the image by an actual area boundary line in a moving picture being played in the second embodiment of the invention.

20 Figures 17A and 17B are an example showing how a locus of motion of the image is modified by an actual area boundary line in a moving picture being played in the second embodiment of the invention.

DETAILED DESCRIPTION OF THE EMBODIMENTS

Now, embodiments of the present invention will be described in detail by referring to the accompanying 25 drawings.

Figure 1 is an example configuration of a computer system used by the composite picture generating

method of the present invention. This system comprises a processor (CPU) 100, a main memory 110, a frame memory 120, a display 130, a key board 140, a pointing device such as a mouse 150, a secondary storage device 160 and a bus 170 to connect them. The main memory 110 is preloaded with a composite picture generating or image synthesizing program 111 and also stores image data 112, image locus data 113, boundary line data 114 and boundary identification information 115, all described later. The secondary storage device 160 stores in advance moving picture data 161 such as real video footage, over which an image is to be superimposed, and moving picture data 162 that was superimposed with an image.

The processor 100 starts the image synthesizing program 111 stored in the main memory 110 to display an image synthesizing program view on the display 130. Then the image synthesizing program 111 operates interactively with the user. First, the moving picture data 161 stored in the secondary storage device 160 is read out and displayed in a preview area 210 (Figure 2) of the image synthesizing program view on the display 130 through a frame memory 120. While watching the moving picture thus displayed, the user manipulates the keyboard 140 or pointing device 150 to enter image data to be synthesized into the moving picture, and specify the locus of motion of the image over the moving picture while playing the moving picture and, if necessary, a boundary line and boundary identification information (e.g., color

information) in the moving picture that restrict the movement of the image. The image data 112 entered by the user is stored in the main memory 110 and the image locus data 113 and the boundary identification information 115 5 are also stored in the main memory 110 in accordance with the specification of the user. Based on the stored image locus data 113 or, if necessary, on the boundary line data 114 and the boundary identification information 115, the image synthesizing program 111 superimposes the image 10 data 112 over the moving picture data 161 while the moving picture data is played, and displays the synthesized image in the preview area of the image synthesizing program view on the display 130. The moving picture data 162 superimposed with the image is stored in 15 the secondary storage device 160, from which it can be retrieved and used as many times as desired.

Figure 2 illustrates an image synthesizing program view 200 displayed on the display 130. In Figure 2, a field 210 is a preview area that plays a moving 20 picture for image synthesizing. The preview area 210 is also used as an area to display the image to be superimposed on the moving picture and a locus of motion of the image. A field 220 is an area indicating the time spent to play the moving picture in the preview area 210. 25 A cursor 221 under the field 220 indicates a currently displayed position in the field 220. A field 222 is an area to specify a playback speed. For example, "1" represents a playback at a normal speed, "1/2" a playback

at half the normal speed (slow playback), and "2" a playback at two times the normal speed. This value can be changed by the keyboard 140 or pointing device 150. A field 230 is an area to specify a kind of line and a 5 color with which to draw an image in the preview area 210. A field 240 is an area (buttons) by which the user can select either drawing of an image, generation/modification of a locus or specification of a superimposing area by using the pointing device 150. A 10 field 250 is to control the moving picture being played in the preview area 210 by using the pointing device 150. Selecting a desired icon can specify the corresponding function, such as moving back to the head position, stopping the playback temporarily, stopping the preview, 15 playing back and moving to the end position. A field 260 is an area to specify a moving picture to be played in the preview area 210 either by directly inputting a file name using the keyboard 140 or by selecting it with the pointing device 150. A field 270 is an area (execute 20 button) to specify the execution of synthesizing by depressing it with the pointing device 150.

First, let us explain about processing, as the first embodiment of the invention, which superimposes an independent image over the moving picture by manually 25 drawing an image and specifying with the pointing device the locus of motion of the image over the moving picture while playing the moving picture.

Figure 3 is a flow chart representing an

overall processing flow of this embodiment. Here it is assumed that the image synthesizing program 111 has already been started and that the image synthesizing program view 200 is displayed as shown in Figure 2.

- 5 Using the pointing device 150 or keyboard 140, the user chooses in the field 260 a file name of a moving picture to be superimposed with an image (step 301). Next, when the user selects a playback start button in the field 250, the corresponding moving picture data 161 is read
- 10 out from the secondary storage device 160 and displayed on the preview area 210 (step 302). The user depresses a stop button in the area 250 at a desired moving picture playback position to temporarily halt the playback of the moving picture, selects the drawing button in the field
- 15 240, selects a kind of line and a color in the field 230, and draws in the preview area 210 an image to be superimposed over the moving picture (step 303) by using the pointing device 150. When the drawn image is determined, the user selects the locus button in the
- 20 field 240, typically resumes the playing of the moving picture at a low speed, draws and determines the locus of motion of the drawn image in the preview area 210 with the pointing device 150 as the moving picture is played (step 304). The image and the locus of motion of the
- 25 image drawn in the preview area 210 are stored in the main memory 110 as the image data 112 and as the image locus data 113 representing position data at predetermined intervals. Finally, when the user

depresses the synthesize execute button in the field 270, the synthesizing of the image data 112 and the moving picture data 161 is executed based on the image locus data 113 (step 305). The synthesizing operation involves 5 pasting the image data 112 successively to the moving picture frames at corresponding positions based on the time data and position data of the image locus data 113 as the moving picture following the temporary stop is played back.

10 Figure 4 is a flow chart showing the detail of the step 304 of Figure 3. This is explained by referring to an example of Figures 5 and 6.

Figure 5 shows an image synthesizing program view 200 with an image drawn by the step 303 of Figure 3. 15 In the preview area 210, the drawn image (goldfish in this case) 500 is superimposed over the moving picture. The field 240 indicates that the drawing button is selected. The cursor 221 under the field 220 shows the point in time at which the moving picture was stopped. 20 The field 222 indicates that the moving picture was being played at the normal speed.

The user selects the locus button in the field 240 and determines the speed at which to play the moving picture (step 401). In this case, the playback speed is 25 set, for example, at half the normal playback speed (slow playback) to ensure that the locus of motion of the image can be drawn easily as the moving picture is played.

Next, the user resumes the playback of the moving picture

by operating a particular key on the keyboard 140 (step 402) and at the same time moves the cursor representing the position of the image 500 in the preview area 210 to draw the locus of motion of the image (step 403). At 5 this time, the speed of the position cursor of the image is adjusted by using the pointing device 150 to match the playback speed of the moving picture. When the drawing of the desired locus of motion of the image is finished, the playback of the moving picture is ended by depressing 10 a particular key on the keyboard 140 (step 404).

Figure 6 shows the image synthesizing program view 200 at this time. In Figure 6, a reference number 600 represents a cursor representing the position of the image and 610 a locus drawn by the step 403. In Figure 15 6, the image 500 of Figure 5 is omitted for the sake of simplicity. The field 240 shows that the locus button is selected. The cursor 221 under the field 220 has moved to a time position 221' (corresponding to the front end of the locus). The playback speed in the field 222 is 20 1/2 (slow playback). The locus 610 is sampled and read in at particular time intervals or frame intervals and the position data of a series of locus points and others are stored as the image locus data 113 in the main memory 110. A frame number from which the playback is resumed 25 by the step 402 and a frame number at which the playback is ended by the step 404 are also stored.

After having drawn the motion locus of the image, the user selects the playback button in the

function selection field 250 to preview and check the locus 610 (step 405) to decide whether or not to modify the locus 610 (step 406). When the user wishes to modify the locus 610, i.e., the motion of the image, the user 5 moves to step 407. If no modification is made, the processing is ended. If, in the step 407, the user decides to redraw the locus from scratch, the user determines the start position for redrawing (step 408) and returns to step 401. When a fine adjustment is to be 10 made, the drawn locus 610 is directly modified (step 409). The redrawing or direct modification is specified by, for example, clicking on the locus button again in the field 240 or by a right-button click.

Next, a method of directly modifying or finely 15 adjusting the locus of motion of the image will be explained by referring to Figures 7 and 8. Figure 7 is an enlarged view of the previewed locus 610. Figure 8 shows a data structure of the image locus data 113 corresponding to the locus 610.

20 When the locus of the image is to be finely adjusted, a part of a series of locus points d1-d9 shown in Figure 7 needs only to be modified (by addition or deletion). In Figure 7, 700 shows a series of locus points before modification, 710 shows a series of locus 25 points with some points, such as a1 and a2, added, and 720 shows a series of locus points with some points such as d8 deleted. Here, the series of locus points represents the positions picked up at one-second

intervals (with the normal playback taken as a reference). It is of course possible to change the point pickup interval to a one-frame interval or 0.5-second interval. A table 800 in Figure 8A corresponds to the 5 before-modification series of points 700, d1-d9, of Figure 7. Here, it is possible not only to specify the locus but also to specify or modify the size (magnification factor), direction and transmittivity of the image to be superimposed. When it is desired to slow 10 down the motion of the image, some points, such as a1 and a2, are added where one wants the image to slow down, as indicated by 710 of Figure 7. As a result, data of a1 and a2 is added as shown in a table 810 of Figure 8B, changing the locus to the one where the image moves 15 slowly between d6 and d7. Conversely, when it is desired to quicken the motion of the image, a point d8 for instance is removed from where one wants the image to move quickly (although in the figure the point d8 to be deleted is shown dotted, it is actually not displayed). 20 As a result, the data corresponding to the point d8 is deleted as shown in the table 820 of Figure 8C, shortening the moving time between the points d7 and d9. Although the locus can be modified easily by manipulating the series of previewed points using the pointing device 25 150, the modification can also be done by reading the image locus data 113 from the main memory 110 and directly changing the corresponding field.

As described above, by moving the pointing

device 150 in the preview area of the image synthesizing program view, the locus of an image can be generated and modified easily.

Although this embodiment has described in the 5 case where the playback speed when the locus is entered is constant, it is also possible to speed up or slow down the playback speed of the moving picture by an input device such as keyboard while at the same time drawing the locus with the pointing device. Further, while the 10 flow chart of this embodiment shown in Figure 3 separates the locus drawing processing from the synthesizing of the moving picture and the superimposed image, it is of course possible to perform these processing simultaneously.

15 Next, a second embodiment of the present invention will be described which specifies a boundary line and area identification information in a moving picture to automatically modify the locus of the image or to automatically modify the position where the image is 20 superimposed over the moving picture.

Figure 9 shows in flow chart an overall processing flow of this embodiment. It is assumed that the image synthesizing program 111 has already been started and the display 130 displays the image 25 synthesizing program view 200 as shown in Figure 2. In Figure 9, steps 901 to 903 are similar to steps 301 to 303 of Figure 3 of the first embodiment. When the image to be superimposed over the moving picture has been

determined at step 903, the user selects the area button in the field 240 to specify, by using the pointing device 150 on the moving picture in the preview area 210, a boundary line and area identification information of an 5 area in which the image is allowed to move (step 904). Next, the user selects the locus button in the field 240 and resumes the playback of the moving image at a low speed, for example, to draw and determine the locus of motion of the image as the moving picture is played (step 10 905). This is basically the same as the step 304 of Figure 3, except that the locus of motion of the image is automatically corrected so that it is within the area specified by the step 904. Finally, when the user depresses the synthesize execute button in the field 270, 15 the image and the moving picture are synthesized (step 906). This, too, is basically the same as the step 305 of Figure 3 except that the position of the image is automatically corrected according to a change in the boundary line in the playback moving picture.

20 Figure 10 is a flow chart showing the detail of the step 904 of Figure 9. Now let us assume that the displayed state of the preview area 210 of the image synthesizing program view 200 is as shown in Figure 11. In Figure 11, reference number 1100 represents an image 25 (ball) drawn by the step 903, 1120 represents an area A or sky, and 1121 represents an area B or ground. The range in which the image 1100 is allowed to move should be limited to the inside of the area A. The user

specifies a boundary line 1130 between the two areas with the pointing device 150 (step 1001). Next, by using the keyboard 140 or pointing device 150 the user selects representative colors and other information of the areas

5 A and B divided by the boundary line 1130 as information for identifying the areas A, B (step 1002). The boundary line and the area identification information specified by the steps 1001 and 1002 are stored in the main memory 110. They are indicated by the boundary line data 114

10 and the boundary identification information 115 in the main memory 110 of Figure 1.

Figure 12 is a flow chart of processing that automatically corrects the locus of the image so that it is within the boundary and area specified by the step 904 of Figure 9. The processing of Figure 12 may be executed simultaneously in synchronism with the drawing of the locus of motion of the image, or performed on the data of the image locus that has already been drawn.

First, the boundary condition for the movement of the image specified by the step 904 of Figure 9 is set (step 1201). Then, for each position of a series of locus points, the processing from step 1203 on down is executed. When an (x, y) coordinate is set on the screen, a check is first made to determine whether a boundary condition is set for the Y coordinate (step 1203). If it is not set, the program moves to step 1027. If it is set, comparison is made between the Y coordinate of a position of interest in a series of the locus points

and the Y coordinate of the boundary condition (step 1204). Then, it is checked whether the boundary condition is met (step 1205). If the boundary condition is met, the program proceeds to step 1207. If not, the Y 5 coordinate of the position of interest is corrected to meet the boundary condition (step 1206) before moving to step 1207. The step 1207 checks whether a boundary condition is set for the X coordinate. If it is not set, the program returns to step 1202. If the boundary 10 condition for the X coordinate is set, the X coordinate of a position of interest in a series of the locus points is compared with the X coordinate of the boundary condition (step 1208). Then, a checks is made to determine whether the boundary conditions is met (step 15 1209). If the boundary condition is met, the program returns to step 1202. If not, the X coordinate of the position of interest is corrected to meet the boundary condition (step 1210) before returning to step 1202. The step 1202 checks whether the series of locus points has 20 all been checked. If not, the processing from step 1203 on down is repeated for the next locus point. If all the locus points are processed, the correction processing is ended. While in Figure 12 the locus points have been described to be processed in the order of Y coordinate 25 and X coordinate for each position in the series of locus points, they can be processed in the reversed order. Alternatively, a Y coordinate group and an X coordinate group for the entire series of locus points may be

processed separately.

Next, by referring to Figures 13, 14 and 15, an example processing of steps 1204-1206 of Figure 12 associated with Y coordinate will be described. From the following explanation the processing of the steps 1208-1210 of Figure 12 associated with X coordinate can easily be understood by analogy.

Figure 13 shows the locus of motion 1310 of the image 1100 of Figure 11 drawn on the screen without considering the boundary line 1130. Figure 14 shows position coordinate data of the boundary line 1130 and Figure 15 shows position data of a series of points in the locus 1310 spaced at one-second intervals. The data shown in Figures 14 and 15 stored in the main memory 110 are actually more detailed data, but only representative positions are shown for the sake of simplicity.

For the series of points on the locus 1310, their Y coordinates plus the radius of the image (ball) 1100 must be higher (smaller) than the boundary line 1130. As shown in Figures 14 and 15, the Y coordinate of point d3 is lower (larger) than the Y coordinate of the boundary line 1130. This does not meet the condition that the image (ball) 1100 must move without crossing the boundary line 1130 (i.e., the image must move as if it is on the boundary line). So, the Y coordinate of point d3 is modified so that the lower part of the image 1100 is on the boundary line 1130. This synthesizes the image 1100 so that it appears to move on the boundary line 1130.

near the point d3.

Figure 16 shows a flow chart of processing which, in the step 906 of Figure 9 for synthesizing the moving picture and the image, automatically modifies the position data of the locus points of the image when the boundary area in the moving picture being played changes.

First, the area boundary line coordinate data and area identification information such as typical colors of areas, specified by the step 904 of Figure 9, are set (step 1601). Then, the position data of the series of locus points of the image (image locus data) is retrieved as the moving picture is played and, for each locus point, the processing from step 1603 on down is executed. First, based on the area identification information, the actual area boundary line coordinate at the present time in the frame of the moving picture being played is determined (step 1603). A difference between the actual area boundary line coordinate and the boundary line coordinate set by the step 1601 is calculated (step 1604). The difference value may be obtained by determining a reference position for the real and specified boundary lines in advance and calculating their difference at the reference position, or by determining the differences between the boundary lines at individual positions and calculating their average. This difference value is added to the coordinate value of the position data for the locus point of interest to modify the position coordinate of the image and synthesize the

moving picture and the image (step 1605) before returning to step 1602. In step 1602, it is checked whether the series of the locus points has all been processed. If all the locus points are not processed, the steps 1603-5 1605 are executed for each of the remaining locus points in each frame of the moving picture being played. When all the points are processed, the image synthesizing processing is ended.

Next, by referring to Figures 17A and 17B, an example processing of step 1605 of Figure 16 will be explained. Figure 17A represents a state before modification and Figure 17B represents a state after modification. Reference number 1130 represents an area boundary line specified by the step 904 of Figure 9 and 15 1710 represents an actual area boundary line in the currently replayed moving picture that is determined by the step 1603 of Figure 16. The value d is the difference between these two boundary lines. The coordinate positions of the image 1100 as shown in Figure 20 17A appears unnatural. Thus, in step 1605 of Figure 16, the position coordinates of the series of locus points are moved upward by the difference value d to obtain a natural synthesized image as shown in Figure 17B.

In the above embodiments, we have described example cases where a handwritten image is moved over the moving image. The image to be superimposed can be not just a handwritten static image but a prepared animation or moving picture. When specifying the locus of an

image, it is possible to specify or modify not only the position but also the size, direction and transmittivity of the image to be superimposed.

In the embodiments the user draws the locus of motion of the image. When an object is moved over the moving picture at a constant speed or at a constant acceleration for example, specifying the initial position, direction and speed (unit movement) of the moving object can automatically determine the locus of motion of the object as the moving picture is played. In this case, the user does not need to draw the locus. Of course, in this case too, it is possible to make a modification similar to that described in the preceding embodiments by specifying the boundary area in which the object is allowed to move and the boundary identification information.

The image synthesizing program for implementing the image synthesizing method of this embodiment can be described in a computer-implemented language and stored 20 in computer-readable recording media, such as floppy disk, CD-ROM and memory card, for transfer.

As described above, this invention allows an arbitrary image with a natural motion to be superimposed over a moving picture without requiring complex computations, making it easy to create an animation in which the image can be made to move in a desired location as by using a pointing device.